Forensic Analysis through Goal-Oriented Logging

Sean Peisert
UC San Diego

research done with Matt Bishop (UC Davis),
Sid Karin (UCSD), and Keith Marzullo (UCSD)

SHERLOCK HOLMES: "It is of the highest importance in the art of detection to be able to recognize out of a number of facts which are incidental and which vital. Otherwise your energy and attention must be dissipated instead of being concentrated."

-Sir Arthur Conan Doyle,
"The Adventure of the Reigate Squire,"
The Strand Magazine (1893)

Introduction

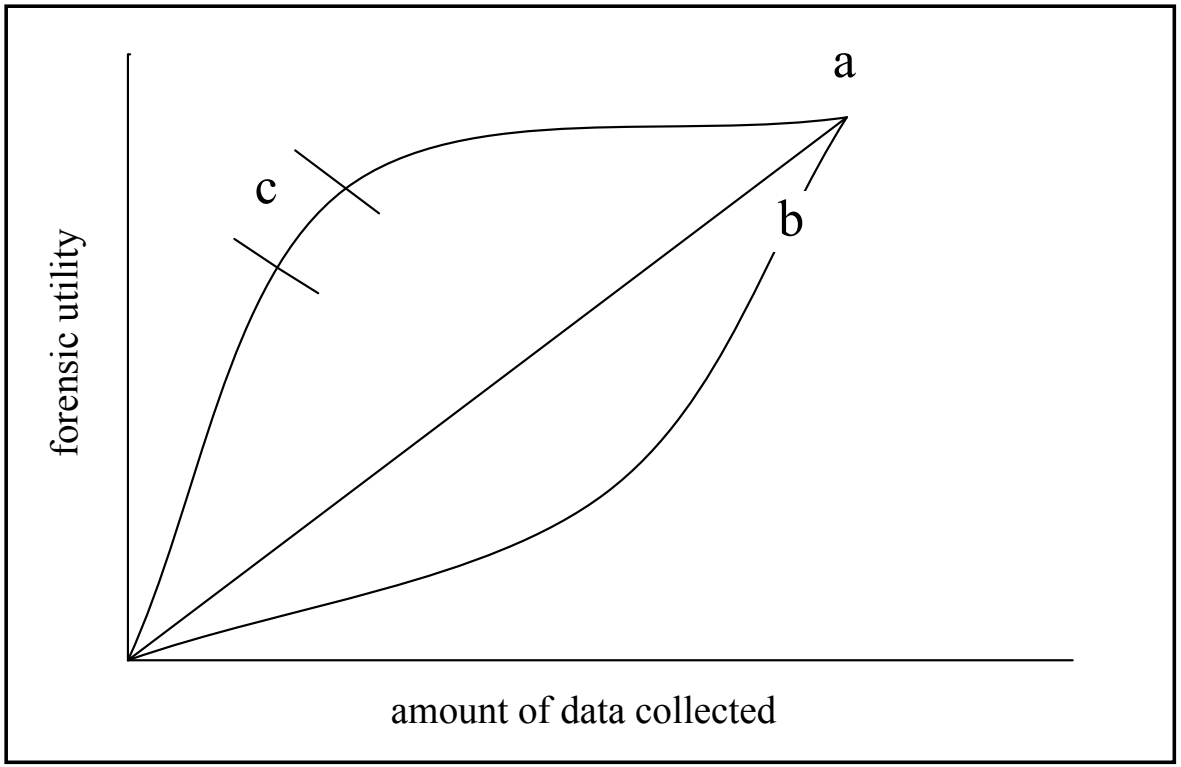
What is forensic analysis?

Forensic analysis vs. intrusion detection

The Problem: Garbage in, Garbage Out

Existing work?

Logged Data vs. Forensic Utility



Existing Work

- Ad Hoc:
  - Focused solutions: Syslog, TCPwrappers, Coroner's Toolkit, Tripwire, LAFS [Wee95], other file auditing [Bishop88]
  - Global solutions: "toolbox approach" [e.g. Farmer & Venema 2004], Sun BSM, BackTracker [King06], Function Call Monitoring [Peisert, Bishop, et al., sub to IEEE TDSC 2006]
- Models: Model of Auditing and Logging [Bishop89], Analysis of Intrusions [Gross97], Model of Security Monitoring [Kuperman04]

Background

Must be a better solution!

Principles of Forensic Analysis
[Peisert, Bishop, et al. in NSPW'05]

Guidelines of a Forensic Model
[Peisert, Bishop, et al., sub. to SADFE'07]

How Do We Do Good Forensics?

- Principle 1: Consider the entire system
- Principle 2: Don't make assumptions about attacks
- Principle 3: Consider effects, not just actions
- Principle 4: Context assists in understanding
- Principle 5: Actions and results must be presented in a way that is analyzable by a human

- But what's in a good forensic model?

Principle 1

Principle 1: Consider the Entire System

Guideline: Indicate the information to log and let the analyst choose whether to record information

Guideline: Provide tuning parameters

Guideline: Automated metrics could help

Principle 2

Principle 2: Don't make assumptions about attacks

- Guideline: Place bounds on unknown stages of attacks

Principle 3

Principle 3: Consider effects, not just actions

Guideline: Consider both pre-conditions and post-conditions

Principle 4

Principle 4: Context assists in understanding

Guideline: Consider the contextual elements surrounding an event, e.g., credentials, IP addresses, environment variables.

Principle 5

- Principle 5: Actions and results must be presented in a way that is analyzable by a human.
  - Guideline: Make the data well-formed [Bishop95]
  - Guideline: Enable association of discrete events to analyze larger attacks [Zhou, et al. 07]
  - Guideline: Make logged events and actual events one-to-one to enable automated translation.

Our Approach

- Builds upon forensic principles & guidelines
- Builds upon formalization of multi-stage attacks
  - [Templeton & Levitt NSPW'00]
  - [Zhou, Carlson, Bishop, et al. TISSEC'07]
- Uses requires/provides model
- [Sub. to Oakland'06; Peisert, Bishop et al.]

Definitions

"attack": sequence of events that violates a security policy (could be internal, as in the insider problem [Bishop & Peisert: UC Davis Tech Report CSE-2006-20])

"goal": to achieve a particular result or violation

"attack graph": Multiple goals liked together in dependency order (related to [Schneier99])

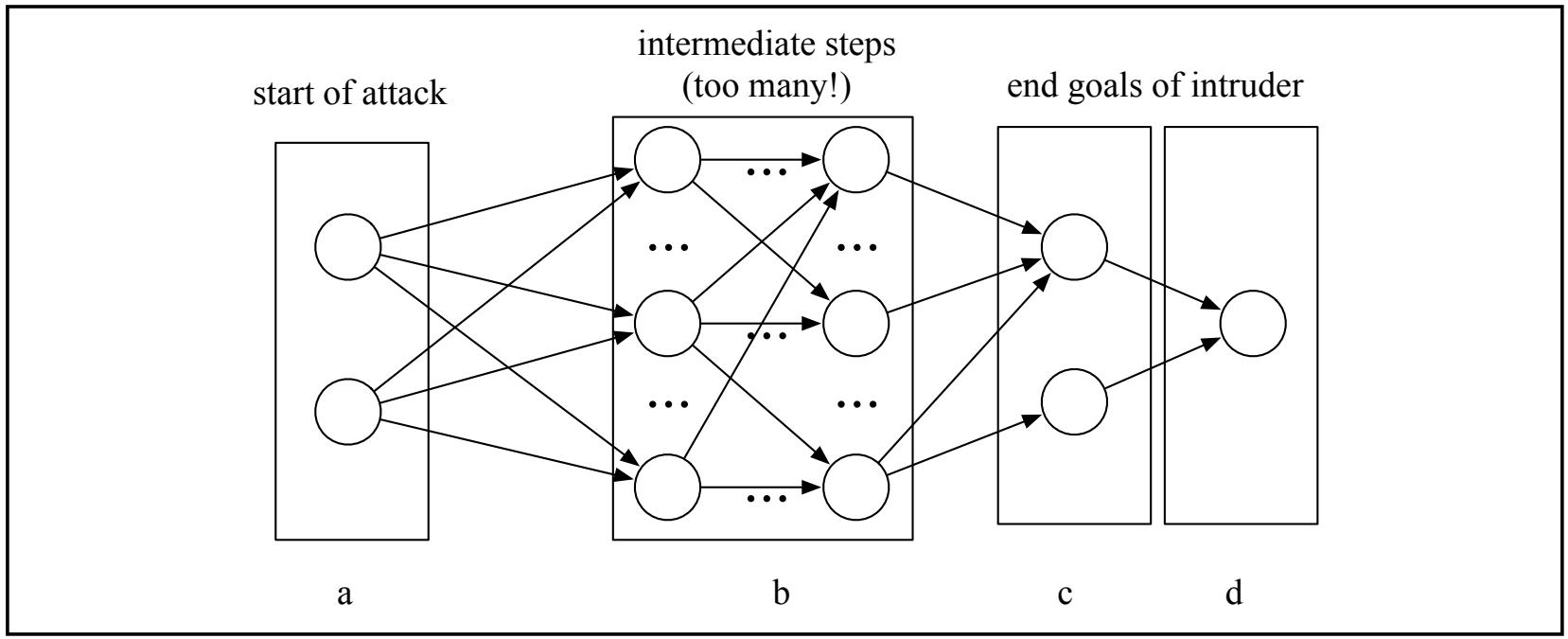
Assumptions

We know an attack/intrusion/something has taken place. (We're analyzing, not detecting.)

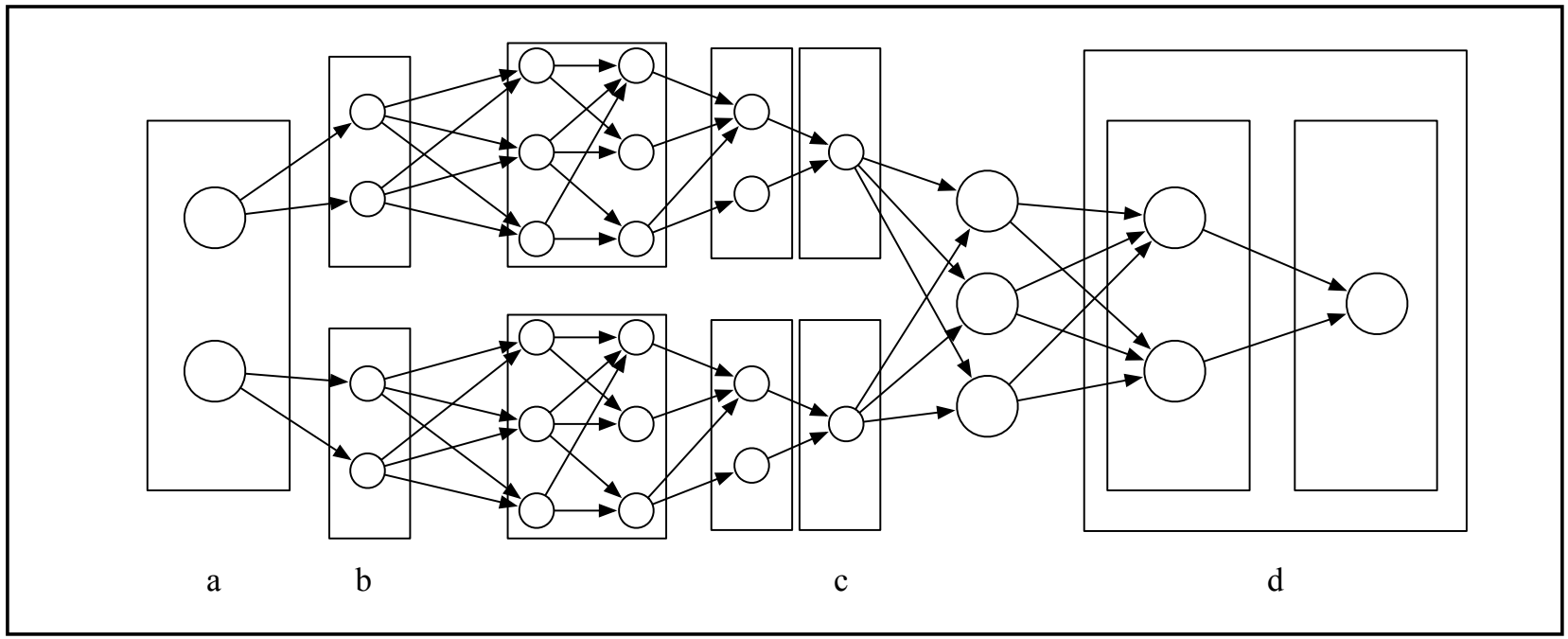
The forensic software obtains accurate information from the system.

The forensic software is able to report this information correctly. [Thompson CACM'84]

Basic Attack Graph



Complex Attack Graph



Methodology

1. Start with an attack graph representing attacker goals to achieve a set of results
2. Work backward from ultimate goal
3. Generate a 6-tuple from each goal
4. Extract information to log from 6-tuple

Premise and Methodology of Building Models, Logging, and Interpreting Data

- Choosing "Goals" and Building Attack Graph
- Building Requires/Provides Capability Pairs
- Extracting Data to Log from the Formalization
- Interpreting Logged Data

Choosing "Goals"

- Currently manual. Eventually:
- Based on policy? [Bishop, Wee, & Frank 1996]
- How to define policy? [Bishop & Peisert, 2006 UC Davis Tech Report]
  - Hard!
  - e.g. "no writes down" - Bell & LaPadula

Building Requires/ Provides Pairs

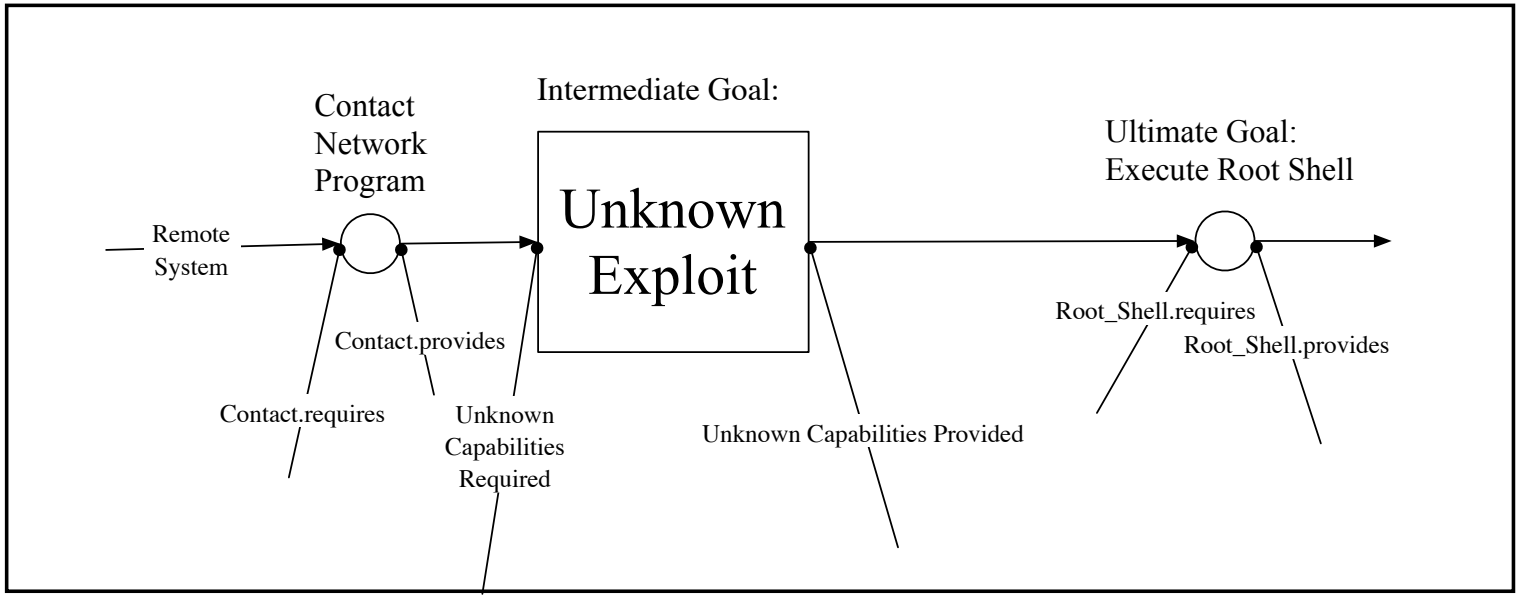
capabilities: a 6-tuple (based on [Zhou07])

- src/dest
- credentials
- actions
- services
- properties

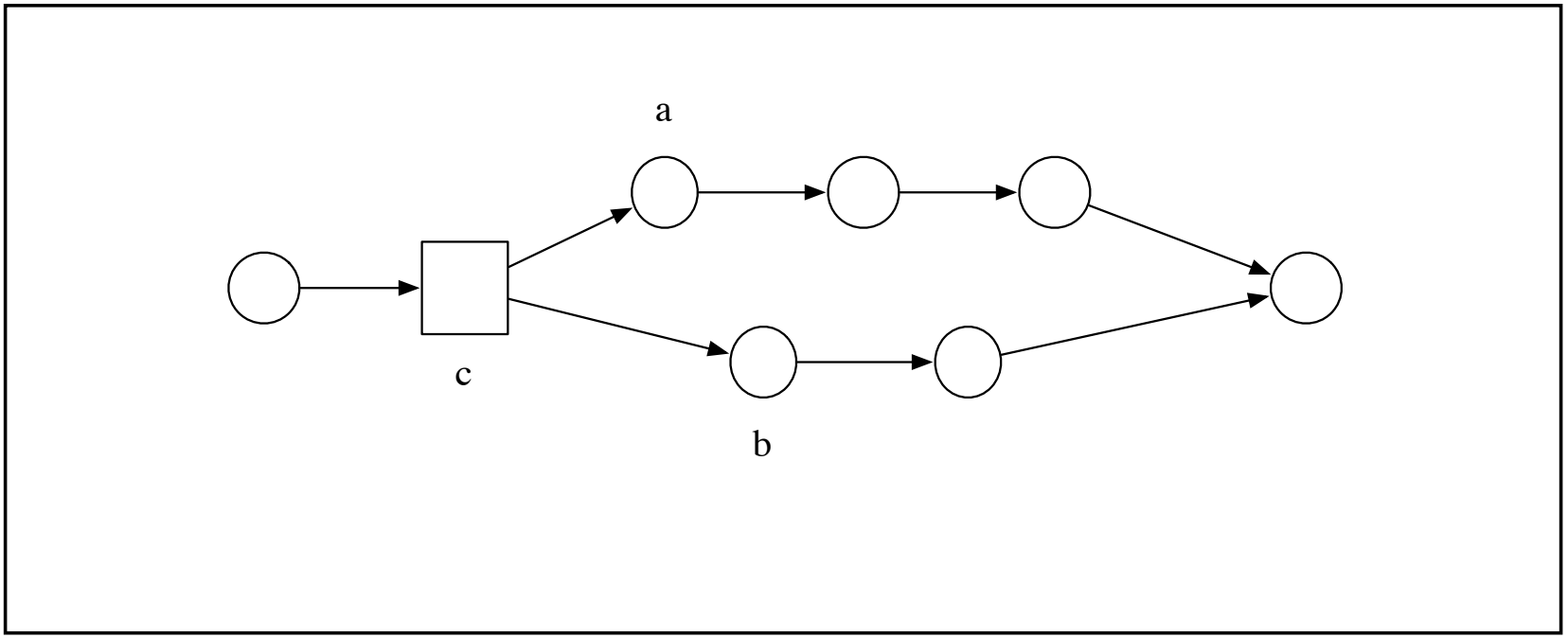
Extracting the Data to Log from the Formalization

- Intuition
- Functions
  - λ : outputs an 8-tuple representing the combined capability pair(s)
  - τ : helps to put bounds on an unknown intermediate step
  - μ : unions together multiple goals
- Algorithms
  - BOUND-UNKNOWNS
  - ANALYZE-ATTACK-GRAPH
  - ANALYZE-GOAL

Applying Tau



BOUND-UNKNOWNS Alg: Applying Tau and Mu



Analysis Algorithms

ANALYZE-ATTACK-Graph

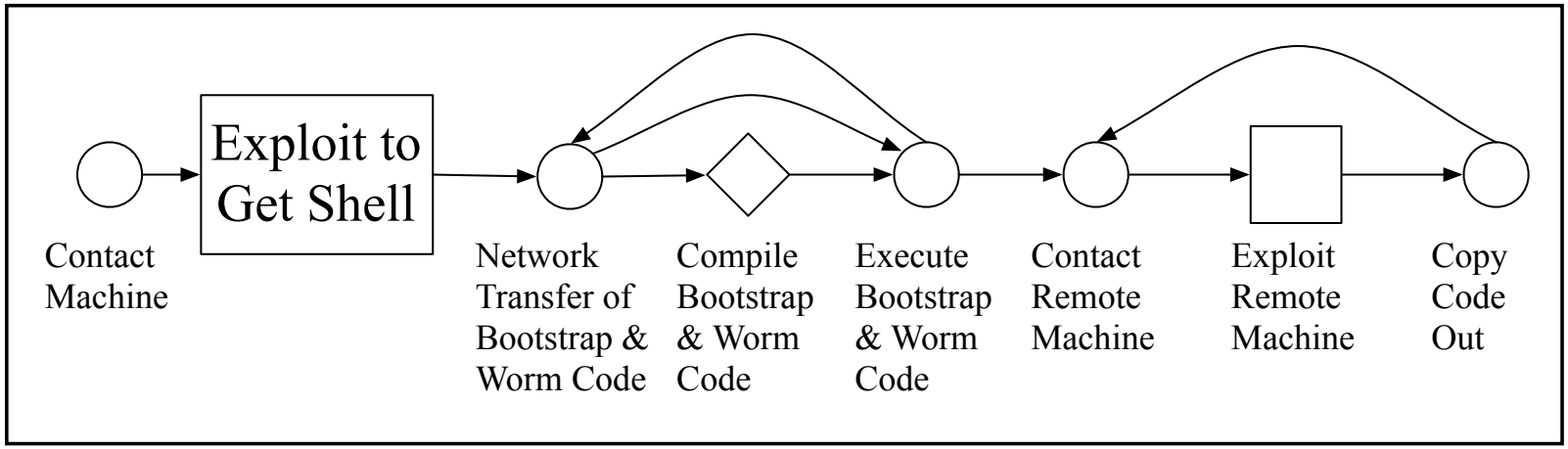
ANALYZE-GOAL

Filter by src, dest, and credential

Determine logging point (e.g. kernel call, hardware) by action, svc., and property

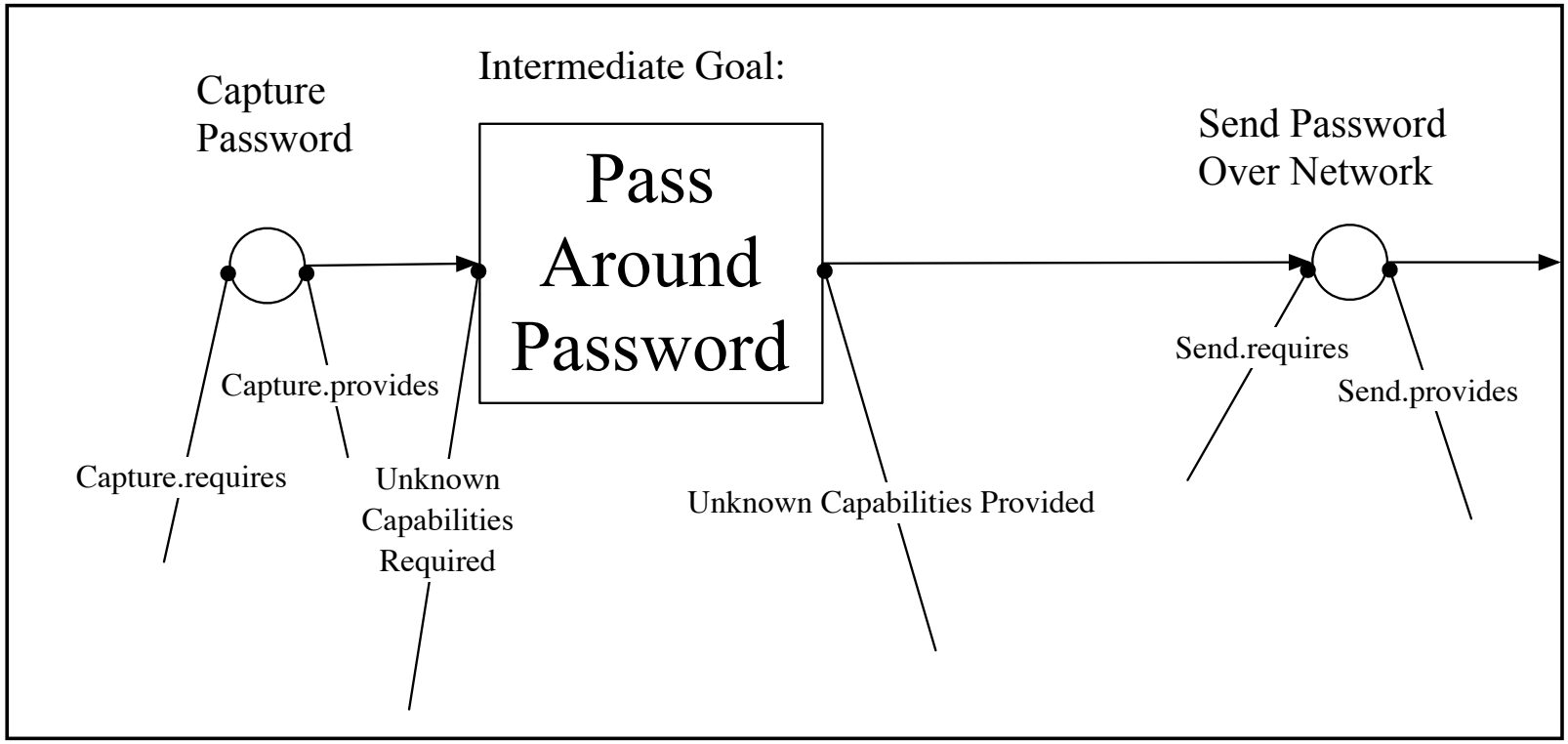
Determine data to log (e.g. syscall, syscall params, assembly code, environment)

Example 1: Morris Worm



[Bishop88, Eichin & Rochlis '89, Seeley89, Spafford89]

Example 2: Spyware



[A. Singer, "Tempting Fate," USENIX ;login:, Feb'05]

Spyware: What to record?

Two things:

- Capturing the password
  - Monitor program that is expected to ask for it.
- Sending the password
  - Opening and use of a socket

Spyware: Modeling Goals

Capture_Password.requires:

{(local, local, ANY(uid:**u**), read, ANY(**PA**) ^ ANY(shell) ^ kernel, password:**P**)}

Capture_Password.provides

{(local, local, ANY(uid:**u**), ∅, Account(P), password:P)}

Send_Password.requires:

{(local, ANY(IP), ANY(uid:**u**), Communicate.send, ANY(Program), password:**P**)}

Send_Password.provides:

{(ANY(IP), local, ∅, Communicate.connect, login(**P**), Account(**P**))}

Implementation Details

FreeBSD 5.x System

Flaws are re-creations of actual exploits
(Spyware example from SDSC intrusion)

Mostly instrumented kernel to get data

Implementation Results of Example 2

Prog	Call	Arg2	Arg4. sa_data	RetVal
ssh	pam_ authenticate			0
ssh	socket			0
ssh	sendto	MyPasswd	192.168.0.1	8

Example 3: lpr Bug

- [8lgm]-Advisory-3.UNIX.lpr.19-Aug-1991
- lpr is/was setuid **root**
- A symbolic link is created from a file to /etc/passwd.
- lpr is called 99 times
- On the 100th time, the first spool file is reused, and with the **-s** argument, lpr follows the symlink to /etc/passwd and copies a specified file to the destination of the symlink, having been running as root.
- Multiple flaws: non-atomic open (**creat**), re-use of spools, etc...

lpr bug: What data to record?

Two steps:

- open() syscall when lpr reads (and accepts) the temp file that we have arbitrarily written
- include symlinks!

lpr bug capability pair

Modify_passwd.requires:

{(local, local, uid=ANY(uid:**u**) ^ euid=root,
Write, file, /etc/passwd)}

Modify_passwd.provides:

{(local, local, user:**u**, Write, file, /etc/passwd
local, local, user:**u**, know, ALL(users), username
local, local, user:**u**, Write, ALL(users), Accounts
)}

Implementation Results of Example 3

Prog	R/Euid	Syscall	Arg1	Arg2	File1 UID	File2 UID
lpr	1001/1001	open	/tmp/.tmp.477	READ	1001	
lpr	1001/0	symlink	/tmp/.tmp.477	/var/spool/ dfA292	1001	0
lpr	1001/1001	rm	/tmp/.tmp.477		1001	
lpr	1001/1001	symlink	/etc/passwd	/tmp/.tmp.477	0	1001
lpr	1001/0	open	/var/spool/ dfA292	WRITE, TRUNC, CREAT	0	

Future Work

- More examples & implementations
- Efficiency measurements & comparisons
- Relative time
- Universal path ID to associate & minimize data.
- Policy Discovery
  - ...to generate attack graphs
  - ...to do automated translation of capability pairs to data necessary to log and where to log it
  - ...to make logged data & events 1:1
  - ...to prove completeness of model

Conclusions

- Forensics is currently ad hoc; a model of forensics is necessary.
- A model needs to be efficient and effective.
- We presented an example of a forensic model ("Laocoön") based on forensic principles.
- Experimental results show that this model of forensic analysis seems to work.
  - Helps to identify and analyze intrusions quickly
  - Mindful of not recording too much or too little data, or just the wrong data.